Creating Dbspaces, Databases, Tables and Indexes in Informix

Create a new dbspace for data called datadbs11 using the steps we used to create the llogdbs01 dbspace. I'll create this dbspace with 1 4GB chunk, feel free to create a smaller or larger dbspace depending on the amount of storage you need/have available. We will use the default page size of 2K, even though different sizes ranging from 2K to 16K are available, this is a topic for another day.

```
informix> touch /home/informix/chunks/DATADBS11.01
informix> chmod 660 /home/informix/chunks/DATADBS11.01
informix> onspaces -c -d datadbs11 -p /home/informix/chunks/DATADBS11.01 -o 0 -s 4194304
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING **  A level 0 archive of Root DBSpace will need to be done.
```

Take a Level 0 backup like we're asked to and verify the dbspace was added by running onstat -d

```
informix> ontape -s -L 0 -d
informix> onstat -d
IBM Informix Dynamic Server Version 14.10.UC7IE -- On-Line -- Up 01:06:04 -- 144148 Kbytes

Dbspaces
address  number   flags     fchunk   nchunks  pgsize   flags   owner    name
4ae5b808 1        0x40001   1        1        2048     N  B    informix rootdbs
4ae5bc80 2        0x40001   2        1        2048     N  B    informix llogdbs01
4be72410 3        0x40001   3        1        2048     N  B    informix datadbs11
 3 active, 2047 maximum

Chunks
address  chunk/dbs     offset     size       free       bpages     flags pathname
4ae5b968 1      1      0          1048576    518077               PO-B- /home/informix/chunks/ROOTDBS.01
4ae5bde0 2      2      0          1048576    11                 PO-B- /home/informix/chunks/LLOGDBS01.01
4be72570 3      3      0          2097152    2097099              PO-B- /home/informix/chunks/DATADBS11.01
 3 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
```

Now lets create an unbuffered logging database named 'blog' in datadbs11 using dbaccess, a curserr based Informix utility for running SQL statements. Typically I would use dbaccess in Menu mode for something like this, but to make the examples clearer I will use dbaccess in Interactive Non-Menu mode (by supplying a hyphen as the second command line argument to dbaccess).

```
informix> dbaccess - -
> create database blog in datadbs11 with log;

Database created.
```

I have successfully created the blog database and am currently connected to it. When I told Informix to create blog in datadbs11 all of the housekeeping stuff for the blog database were put in datadbs11 and datadbs11 will be the default dbspace for any tables, indexes, etc. I create.

Now lets **create some tables**. How about some tables to record blog entries for multiple blogs, call them blog and blog_post. For this type of stuff, things that involve more typing and are a little more complicated, I like to have dbaccess execute a .sql file that I prepare before hand.

```
informix> vi create_tabs.sql

create table blog (
   id            serial not null,
   name          varchar(255)
```

```
) in datadbs11 extent size 256 next size 256 lock mode row;

create table blog_post (
   id            serial not null,
   blog_id       integer not null,
   title         varchar(255)
) in datadbs11 extent size 1024 next size 1024 lock mode row;

informix> dbaccess blog create_tabs.sql

Database selected.


Table created.


Table created.


Database closed.
```

If you're familiar with databases, create_tabs.sql should look familiar enough except for the Informix specific stuff. The Serial data type is an auto-incrementing integer, in datadbs11 puts the table in the datadbs11 dbspace, extent size and next size specify the first and next extent sizes in KB and finally lock mode row enables row level locking on this table vs. page level locking.

Looks like we need some referential constraints and some indexes on those tables. You could do this as part of the table creation, but if you do Informix will name the supporting indexes all funky and if you have to do something with them later it will be a pain. Because of this I like to manually create the indexes and create foreign and primary keys afterwards.

```
informix> vi create_idx.sql

create unique index blog_pk on blog (id) in datadbs11;
alter table blog add constraint primary key (id)
    constraint blog_pk;

create unique index blog_post_pk on blog_post (id) in datadbs11;
alter table blog_post add constraint primary key (id)
    constraint blog_post_pk;

create index blog_post_fk1 on blog_post (blog_id) in datadbs11;
alter table blog_post add constraint foreign key (blog_id)
    references blog (id)
    constraint blog_post_fk1;

informix> dbaccess blog create_idx.sql

Database selected.


Index created.


Table  altered.


Index created.


Table  altered.


Index created.


Table altered.
```

```
Database closed.
```

I created 2 unique indexes with 2 primary keys on top of them and 1 non unique index with a foreign key on top of it. This way requires a little bit more typing, but it is how I like to do it and if you want to put the underlying referential constraint indexes in a non default dbspace and give these indexes a name then you should do it this way too.

Time to put some data in these tables. I'm going to use a INSERT INTO ... VALUES SQL to put data into the blog table and use a dbaccess LOAD FROM .. INSERT INTO SQL to load data from a pipe delimited file into blog_post.

```
informix> vi load.sql

insert into blog (id, name) values (0, "Informix DBA");
insert into blog (id, name) values (0, "Informix technology");

select * from blog;

informix> dbaccess blog load.sql

Database selected.


1 row(s) inserted.


1 row(s) inserted.




id   1
name  Informix DBA

id   2
name  Informix technology

2 row(s) retrieved.


Database closed.

informix> vi load.unl

0|1|Creating Dbspaces, Databases, Tables and Indexes in Informix|
0|1|ZOMG, FYI - IM Informix Tech Support FTW|
0|1|Informix Backup and Restore - The Bare Minimum|
0|2|A bug can undermine your troubleshooting|
0|2|Informix Editions revisited|
0|2|New Informix editions: Bargain time?|

informix> dbaccess blog -

Database selected.

> load from load.unl insert into blog_post;

6 row(s) loaded.

> select * from blog_post;

id       1
blog_id  1
title   Creating Dbspaces, Databases, Tables and Indexes in Informix

id       2
blog_id  1
title   ZOMG, FYI - IM Informix Tech Support FTW

id       3
blog_id  1
title   Informix Backup and Restore - The Bare Minimum
```

```
id      4
blog_id  2
title  A bug can undermine your troubleshooting

id      5
blog_id  2
title  Informix Editions revisited

id      6
blog_id  2
title  New Informix editions: Bargain time?

6 row(s) retrieved.
```

The last thing I want to do is show you how to add a chunk to an existing dbspace, something you might want to do if you need more space for your data. This is very similar to creating a dbspace and is done via the same onspaces command

```
informix> touch /home/informix/chunks/DATADBS11.02
informix> chmod 660 /home/informix/chunks/DATADBS11.02
informix> onspaces -a datadbs11 -p /home/informix/chunks/DATADBS11.02 -o 0 -s 4194304
Verifying physical disk space, please wait ...
Chunk successfully added.

informix> onstat -d

IBM Informix Dynamic Server Version 14.10.UC7IE -- On-Line -- Up 00:26:37 -- 144148 Kbytes

Dbspaces
address  number   flags     fchunk   nchunks  pgsize   flags   owner    name
4ae5b808 1        0x60001   1        1        2048     N  B    informix rootdbs
4ae5bb88 2        0x40001   2        1        2048     N  B    informix llogdbs01
4bd65d28 3        0x60001   3        2        2048     N  B    informix datadbs11
 3 active, 2047 maximum

Chunks
address  chunk/dbs     offset     size       free       bpages     flags pathname
4ae5b968 1      1      0          1048576    518259               PO-B- /home/informix/chunks/ROOTDBS.01
4ae5bce8 2      2      0          1048576    11                 PO-B- /home/informix/chunks/LLOGDBS01.01
4bb51cc8 3      3      0          2097152    2095214              PO-B- /home/informix/chunks/DATADBS11.01
4bdb73e0 4      3      0          2097152    2097149              PO-B- /home/informix/chunks/DATADBS11.02
 4 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always
```